BBN SYSTEMS AND TECHNOLOGIES
A Division of Bolt Beranek and Newman Inc.

ARPA Order Number 7697
Contract Number: N00014-91-C-0115
Contract Duration: 25 March 1991 - 30 June 1995
Principal Investigators: J. Makhoul, (617)873-332
R. Schwartz, (617)873-3360

Final Report

ROBUST CONTINUOUS SPEECH RECOGNITION

Anastasios Anastasakos, Chris Lapre, Francis Kubala, John Makhoul, Long Nguyen, Richard Schwartz, George Zavaliagkos

September 1994

DTIC
ELECTE
SEP 26 1995
B

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC QUALITY INSPECTED 5

19950922 061

ARPA Order Number 7697
Contract Number: N00014-91-C-0115
Contract Duration: 25 March 1991 - 30 June 1995
Principal Investigators: J. Makhoul, (617)873-332
R. Schwartz, (617)873-3360

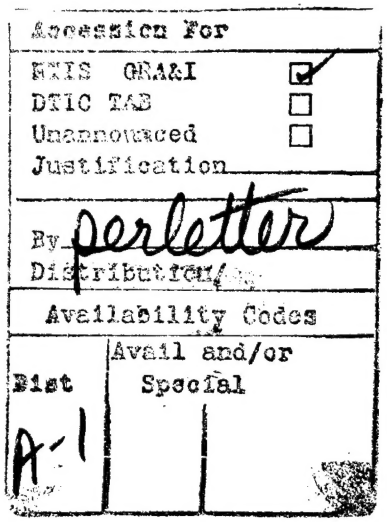
Final Report

ROBUST CONTINUOUS SPEECH RECOGNITION

Anastasios Anastasakos, Chris Lapre, Francis Kubala, John Makhoul, Long Nguyen, Richard Schwartz, George Zavaliagkos

September 1994

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects agency or the U.S. Government.



Contents

Chapter 1

Executive Summary

This is the final report for the project Research in Continuous Speech Recognition, sponsored by the Advanced Research Agency (ARPA) and monitored by ONR under Contract No. N00014-91-C-0115. The report covers the period 25 March 1991 to 30 June 1994.

The objective of this basic research is to develop accurate and detailed mathematical models of the fundamental units of speech (phonemes) for large-vocabulary continuous speech recognition. The important goals of this work are to achieve the highest possible word recognition accuracy in continuous speech and to develop methods for the rapid adaptation of phonetic models to the voice of a new speaker.

The research during the past three years can be categorized into four broad topics: developing better speech and language models to improve the accuracy of large-vocabulary, speaker-independent, continuous speech recognition; developing methods for using these more complex models of speech and language efficiently; developing techniques for dealing with the phenomena found in spontaneous speech; and developing techniques for dealing with differences (along many dimensions, including speaker, accent, microphone, domain, etc.) between the training and test conditions.

In addition to the research performed, a substantial portion of our effort was devoted to the development of speech and language corpora and testing methodologies within the research community which will support and encourage rapid advancement in all of these areas.

The past three years has seen a dramatic change in the problems we have worked on. Prior to this period, the primary focus within the ARPA speech recognition community was on the Resource Management (RM) task, which was a 1000-word read-speech task with a

rather restricted language model. While tremendous progress in acoustic modeling was made on that task, there were many dimensions left untouched. In this project we worked on the recognition of goal-directed spontaneous speech, in the Airline Travel Information Service (ATIS) domain, with a vocabulary of a few thousand words, and the very-large vocabulary Wall Street Journal (WSJ) dictation task. At the same time that we have been learning to deal with the new problems and large sizes of these corpora, we have also seen several further improvements in accuracy.

The advances achieved in this project have been made within BYBLOS, BBN's research system for continuous speech recognition. Among the salient accomplishments, we have:

1. Reduced the word error rate on the ATIS task from 18% to only 3%.
2. After modifying BYBLOS to deal with much larger vocabularies and larger amounts of training data, reduced the word error rate on the WSJ dictation task from 16% to 6% with a vocabulary of 5,000 words, and from over 20% to 12% with a vocabulary of 20,000 words.
3. In 1991, demonstrated the first real-time continuous speech recognition with a vocabulary of over 1,000 words.
4. In 1992, demonstrated the first *software-only*, real-time continuous speech recognition using standard off-the-shelf workstations with no accelerator boards.
5. Demonstrated the first software-only, real-time continuous speech recognition with a vocabulary of 20,000 words in 1993, and with a vocabulary of 40,000 words in 1994.
6. Developed a new methodology for speaker-independent training that requires training speech from only a few speakers.
7. Demonstrated that a system trained only on speech recorded with a high-quality microphone can achieve comparable recognition accuracy when tested on telephone speech.
8. Using rapid speaker adaptation techniques, showed a factor of two reduction in word error rate for nonnative speakers when tested with a system trained only on native speakers of American English.

For the last two years, Mr. Francis Kubala from BBN chaired the CSR (Continuous Speech Recognition) Corpus Coordination Committee. We participated heavily in developing the new "hub and spokes" paradigm for the evaluation of CSR systems, in which all systems are evaluated with a small hub condition, and different systems participate in the different

spokes based on the work being done at each site. The new evaluation procedure is the most comprehensive ever established.

We participated in the annual ARPA Human Language Technology workshops by presenting papers and giving demonstrations of the technology developed under this effort. References to the papers from these workshops, as well as other presentations and papers, can be found in the bibliography: [1, 2, 3, 4, 5, 6, 11, 14, 16, 22, 23, 32, 33, 34, 35, 36, 37].

The research and other work performed under each of these areas is summarized below. In Chapter 2, we describe methods for improved speech modeling. Chapter 3 discusses issues related to training scenarios that make more efficient use of time or speech data. In Chapter 4 we describe improvements to the language models used for speech recognition. And Chapter 5 deals with our work on recognition of spontaneous goal-directed speech, as in the ATIS domain. Finally, we summarize our work on making speech recognition robust to changes in channel, environment, and domain in Chapter 6.

Chapter 2

Improved Speech Models

The primary goal of this project has been to develop more powerful models of speech and language and to develop practical methodologies for training and testing under different conditions. We have considered several different variations in training scenarios. We have also developed methods for dealing with different cases of differences between the training and testing conditions.

In this chapter, we describe the version of BYBLOS used in November '92 for the experiments on training issues. We also describe the improvements made to the system used for the tests performed in November '93. Following this we describe several experiments related to various training issues.

2.1 Improvements to BYBLOS

We extended the BYBLOS system in several ways in an attempt to provide more detailed acoustic-phonetic information and more reliable operation, such as the refinement of robust smoothing techniques and the elimination of extraneous silence periods. We avoid training on passages of speech for which we don't have a reliable transcription, for example, when the speaker stutters badly. We also use various multiple-pass search strategies in order to make the computation and storage manageable for large vocabulary problems, and to be able to incorporate many different knowledge sources efficiently. Most of the computation is performed in parallel on several workstations using a load-balancing batch queue. This allows us to complete fairly expensive training and decoding operations with a short turnaround time.

The November '92 BYBLOS system used 45 cepstrum/energy features per frame, including 14 mel-frequency cepstral coefficients (MFCCs), and their first and second derivatives. We subtract the mean MFCC (measured over the whole utterance) from each frame in the utterance. We use Tied Gaussian mixture (top 5) densities, estimated directly from the initial VQ codebook, and not reestimated during forward-backward training.

We estimate four types of context-dependent phonetic models: triphone, left, right, and context-independent. The lower order models are used both to smooth the triphone models and to synthesize triphone models for those triphones not observed in the training, but required for decoding. We use triphone cooccurrence smoothing [13] to smooth the mixture weights of the context-dependent models. This made it possible to benefit from triphones with only one training sample. We also remove long pauses in the speech using a causal energy-based speech detector. The remaining short pauses in the training speech are detected using a constrained segmentation.

The decoder used a 4-pass search strategy [6]. The strategy used a forward pass followed by a backward Word-Dependent N-best search algorithm [14] with a bigram language model, within-word triphone models, and top-1 (discrete VQ) densities. The N-best hypotheses were then rescored using cross-word context models and top-5 mixture densities, and the appropriate (bigram or trigram) language model for the test being performed.

We used gender-dependent models. To accomodate some speakers whose speech was not modeled well by either the male or female models, we included a third gender-independent model. The answer that produced the highest total score was selected.

2.1.1 New HMM topologies

During the past year, the size of the training corpus of speech was increased by a factor of 5. In order to be able to make use of more training data, we increased the complexity of the phonetic models. First, we increased the model from 3 states to 5. We also experimented with a phonetic model consisting of 13 states in three parallel paths – a short, medium, and long one.

2.1.2 Clustered Densities

We implemented the state clustering algorithm proposed by Hwang [8] using a bottom-up agglomerative clustering technique. The clustering was constrained to group only those states that were in the same position within the same phone. The number of independent

states was reduced by about a factor of five. We did not use the top-down decision tree technique for predicting unseen triphones [9]. Triphones needed for recognition that did not occur during training were synthesized from the appropriate left- and right-context models.

2.1.3 Lattice Decoder

It has frequently been asserted that the N-best decoding strategy must cause search errors when the error rate is high and the sentences are long, as is true with sentences in the WSJ corpus. Therefore, we developed a new 6-pass decoding strategy that combines several multi-pass search paradigms:

1. A fast match algorithm that finds the most likely word-ending scores and times at each frame [5].
2. A backward search using bigram language models, within-word triphones, and discrete HMM models.
3. A forward search using the same models as pass 2.
4. From passes 2 and 3, we construct a lattice of all likely word sequences [10]. The lattice is expanded to allow scoring with trigram language models and cross-word triphone acoustic models. Because there is a separate copy of each word depending on the following word, a biproduct of this pass is the Word-Dependent N-best hypotheses [14].
5. The N-best hypotheses are then rescored independently with any available models, such as 13-state HMM models, a trigram language model, and Segmental Neural Nets (SNN), [15] or Stochastic Segment Models [11].
6. Using weights designed to minimize the word error on a development set, the scores are combined and the hypotheses are reordered to produce a new list. At this point we can choose the top answer or pass the list on to an understanding system for further selection based on meaning.

While it may appear that a 6-pass strategy must be slow, it is not so. The first pass (fast match) is very fast and each successive pass is faster than the previous one, due to the greatly reduced search space. However, we found that the accuracy was only slightly improved over the previous year's strategy, in which the N-best search was performed using bigram, non-crossword models. This shows that there were few search errors in the previous method.

2.1.4 Segmental Neural Networks

In an effort to overcome the independence assumption inherent in HMM models, we have used a Segmental Neural Network (SNN) [15] to model a whole phonetic segment as a single unit. The SNN transforms each variable-length input segment into a fixed-length in order to model all of the parameters jointly. The output of the SNN is a posterior probability for each hypothesized phoneme in each hypothesized sentence. The output scores and the *a priori* probabilities are given as additional scores for each of the N-best hypotheses supplied by the decoder. These scores are then used in combination with the HMM scores.

Each frame of input contained 14 Mel-Scaled cepstra, normalized energy, and delta energy. First we converted the parameters of each variable-length phonetic segment into a fixed-length input by performing a DCT on each input cepstrum sequence, resulting in 5 (smoothed) parameters for each original sequence. The resulting (80) parameters were input to a 2-layer perceptron network. The first layer consisted of randomly chosen weights from the input to 500 hidden units. The second layer had weights from the hidden layer to each of the possible (56) output phones. The second layer was estimated using linear least squares techniques. There was a separate second layer for each possible left or right phonetic context.

This is the first time we used the SNN models for very large vocabulary speech recognition. The use of the SNN typically reduces the word error rate by about 8-10%.

2.1.5 Weight Optimization

In addition to the large number of acoustic and language model parameters in a recognition system, there are several system parameters that must be tuned for optimal performance. Many of these cannot be estimated directly using the same techniques (e.g., maximum likelihood). Some examples of these parameters are: word and phoneme insertion penalties, the grammar weight, the codebook weights, and the weights of alternate acoustic models. The word insertion penalty is an additional probability that we multiply by for each transition to a new word (in addition to the grammar probability). This is used to control the balance between insertions and deletions. The language model weight is an exponent on each grammar probability that allows us to obtain the best balance between the acoustic model scores and the language model scores. Finally, each different acoustic model is weighted by exponentiating the probabilities according to their relative power.

Clearly we would like to set these parameters to optimize recognition accuracy directly. However, maximum likelihood estimation techniques cannot be used to estimate these ex-

ponent parameters. Therefore, we typically run several recognition experiments – each requiring a few hours – to try to find the best system parameters. However, this tuning often requires extensive experience and too many experiments.

The total probability of a sentence hypothesis can be expressed as the product of the exponentiated probabilities of each knowledge source (KS):

$$\begin{aligned}\text{Utt-score} = &\text{HMMScore}^{\alpha} * \\ &\text{GrammarScore}^{\beta} * \\ &\text{WordPenalty}^{\#words} * \\ &\text{PhonePenalty}^{\#phones}\end{aligned}$$

The unknown values are the exponents α and β, and the WordPenalty, and PhonePenalty. If we take the log, we have

$$\begin{aligned}\log \text{Utt-score} = &\log \text{HMMScore} * \alpha + \\ &\log \text{GrammarScore} * \beta + \\ &\#words * \log \text{WordPenalty} + \\ &\#phones * \log \text{PhonePenalty}\end{aligned}$$

Now, the unknown values on the right are just linear weights for the KSs on the left. Admittedly the number of words and phones are simple KSs, but we find that including these terms significantly improves recognition accuracy. We need to find the four values that minimize the error rate. While minimizing error rate directly for continuous speech is usually difficult, it becomes easy if we change the problem to one of minimizing the error rate when choosing among the N-Best alternatives for an utterance. First, we find the N-Best hypotheses for all of the utterances in a development test set. The rescoring step provides the log probabilities for each hypothesis for each KS separately. We use a gradient search to find the set of weights that, averaged over the development set, brings to the top the answer with the smallest number of errors. To evaluate a particular set of weights we compute the total weighted log score for each hypothesis (the dot product of the weights and scores), and then find the hypothesis with the maximum total score for each utterance. We measure the word error rate for this top choice for each utterance in the set in the usual way. The total word error rate over the set is our evaluation function for this set of weights. The computation needed to evaluate a set of weights for 100 hypotheses for 300 test utterances can be measured in milliseconds. Therefore we can consider several thousand weight vectors in a few seconds in our search for the set of weights that minimizes word error rate on the development set. As long as the development test set contains enough utterances – say 300 – we find that the weights found are also good for new test sets.

2.2 Continuous Densities

In the past, we have primarily used discrete-density HMMs and tied-mixture (TM) HMMs. We had found that the accuracy was just as high as for continuous density HMMs, and that the programs required far less computation. However, there is increasing evidence that, as the amount of training speech increases, the accuracy of continuous density HMMs continues to improve, while the tied-mixture HMMs does not improve very much after several hours of training. Therefore, we have also begun to explore the use of HMMs with more densities.

Rather than using completely independent Gaussian density mixtures for each state, however, we have decided to continue to use "codebook-like" densities, with sets of Gaussians tied across several states. This method has been called phoneme-tied mixtures (PTM) when the codebooks are common to each phoneme, or "genones" when the codebooks are more specific [12]. (We prefer to use a more descriptive term for the latter system, which we would call State-Tied-Mixtures.)

The primary advantage of using more codebooks is that it allows us to model all of the 45 parameters jointly within each density. At this point, we have implemented the PTM system with a single parameter stream and have found that the accuracy is roughly the same as we had with our TM system with multiple parameter streams. Thus, the advantage for more codebooks and being able to model dependence is offset by the decreased resolution resulting from using one parameter stream.

We are currently increasing the number of codebooks to about 1,000 based on the state-clustering algorithm.

Chapter 3

HMM Training Issues

One of the important design considerations in developing a speech recognition system is how the system will be trained. We have considered and compared several alternative training scenarios and methods for large vocabulary continuous speech recognition. The Wall Street Journal (WSJ) corpus provides the opportunity to test many of these methods.

We performed five key experiments that were designed to answer questions related to different training scenarios. We investigated 1) the effect of varying the number of training speakers if the total amount of training data remains constant, 2) data pooling versus model averaging for generating Speaker-Independent (SI) HMMs, 3) the benefit of doubling the acoustic training data, 4) SI versus SD performance when the SI training data is twelve times greater, and 5) the effect of cross-domain training for both the acoustic and language models.

3.1 Effect of Number of Training Speakers

It has always been assumed that for speaker independent recognition to work well, we must train the system on as many speakers as possible. For the first time we were able to perform a well-controlled experiment to answer this question on the WSJ0 corpus. We found, to our surprise, that there is little advantage for having more speakers if the total amount of speech is fixed. Specifically, training with 600 sentences each from 12 speakers gave almost the same performance as training with 7,200 sentences from 84 different speakers. The 12 speakers were selected randomly, without any effort to be sure that they covered the general population. In addition, we found that we could also achieve essentially the same

performance if we trained the system separately on each of the speakers and averaged the resulting models, as if we trained jointly on all of the speakers together. Both of these results have important implications for practical speech corpus collection.

The following results were obtained on the 5K VP closed-vocabulary development test set of the WSJ0 corpus using the standard bigram grammar. The experiment was repeated

	Pooled	Averaged
SI-12	11.6	12.0
SI-84	11.2	12.3

on the 5K NVP closed-vocabulary development test and we found the differences to be even smaller between the SI-84 pooled and SI-12 averaged approaches.

3.2 Effect of Additional Training

We usually observe that the recognition error rate decreases as the square root of the amount of training speech. For example, when we double the amount of training, we observe that the error rate decreases by 30%. We doubled the amount of acoustic training by combining the two corpora above resulting in 14,400 total utterances from WSJ0. Results for the 5K NVP development test are shown below. The results show a 5% decrease in error instead of the

	Training Corpus	Word Error
(1)	SI-12, averaged	11.3
(2)	SI-84, pooled	11.2
(3)	SI-12+84, avg of 1 + 2	10.6

expected 30% decrease. This shows that the models used are not able to take advantage of additional training data. Later on, when we used 37,000 sentences of training from the WSJ1 corpus, we also observed a rather small decrease in error rate. We are now improving our overall modeling strategy so that we will be able to take better advantage of more training data.

3.3 Speaker-Dependent vs Speaker-Independent

Below we compare the recognition error rate between SI and SD recognition. The SI models were trained with 7,200 sentences, while the SD models were trained with only 600 sentences each. The SI results were obtained for two different sets of test speakers. For the SD case, we compare two different test sets from the training speakers. These experiments were performed using the 5K-word NVP closed-vocabulary test sets, using the standard bigram language models. As can be seen, the word error rate for the SI model is only somewhat

Training →	SI-12	SD-1
Test	(7200)	(600)
Dev. Test	10.9	7.9
Nov. '92 Eval	8.7	8.2

higher than for the SD model, depending on which SI test set is used. We estimate that, on the average, if the amount of SD training were reduced such that the training speech for the SI model were 15-20 times that used for the SD model, then the average word error rate would be about the same.

3.4 Domain Independence

Hon [7] has shown that, with some care, and large amounts of training speech from several different domains under similar environmental conditions, it is possible to achieve reasonable accuracy on a new domain, given the correct language model and vocabulary.

We performed an experiment in which we compared the effects of deriving the acoustic training data, the vocabulary, and the language model statistics from other domains. Our test set was the Feb. 1992 ATIS test set. The other training was limited to the WSJ0 corpus. The ATIS2 training set contains approximately 10 hours of spontaneous speech training, while the WSJ0 SI-12 training corpus contains 12 hours of read speech. For these experiments we used a bigram grammar only, with cross-word rescoring of the acoustic models. When the grammar was from ATIS, the vocabulary contained 1830 words, including about 400 ATIS-specific compound words. The "ATIS null" grammar assumes all words are equally likely.

The table below shows, for each condition, the language model (which includes the

vocabulary), the source of the acoustic training, the resulting word error rate, and the error factor, which is the ratio of the word error rate to the control condition. From these

Language Model	Acoustic Training	Word Error	Error Factor	
ATIS bigram	ATIS	9.2	1.0	(Control)
ATIS bigram	WSJ	13.0	1.4	
ATIS null	ATIS	28.6	3.1	
ATIS null	WSJ	50.5	5.5	
WSJ 20K bigram	WSJ	55.3	6.0	

preliminary experiments one can derive two obvious conclusions:

1) Given a reasonably diverse acoustic training set, there is relatively little degradation in the acoustic model when moving to a new domain. This was true, despite the fact that the amount of training was approximately equal in both cases above, and also that the training data was read speech, while the test was spontaneous speech. Furthermore, the training data was collected at two sites, whereas the ATIS test data originated from 5 sites.

2) The lack of an appropriate language model is clearly a much more serious problem. Whether we use a null grammar with the correct vocabulary, or a well-trained language model from another source, the error rate is degenerate.

3.5 Summary of Results

We summarize our experiments on training issues as follows:

1. Counter to intuition, given a fixed amount of training speech, increasing the number of training speakers has little effect on performance. In addition, averaging independently trained SD models is nearly as good as pooling the training data to estimate the maximum likelihood models.

2. Our current models are not able to utilize large amounts of acoustic training material to improve performance.

3. Speaker-independent (SI) recognition with 12 hours of training speech is almost as good as speaker-dependent (SD) recognition with one hour of training.

Chapter 4

Language Modeling Improvements

Speech recognition accuracy is affected as much by the language model as by the acoustic model. In general, for language models of the same type, the word error rate is roughly proportional to the square root of the perplexity of the language model. In addition, in a natural unlimited vocabulary task, a substantial portion of the word errors come from words that are not even in the recognition vocabulary. These out-of-vocabulary (OOV) words have no chance of being recognized correctly. Thus, our goal is to estimate a good language model from the available training text, and to determine a vocabulary that is likely to cover the test vocabulary.

The straightforward solution to improving the language model might be to increase the complexity of the model (e.g., use a higher order Markov chain) and/or obtain more language model training text. But this by itself will not necessarily provide a better model, especially if the text is not an ideal model of what people will actually say. The simple solution to increase the coverage of the vocabulary is to increase the vocabulary size. But this may also increase the word error rate and it increases the computation and size of the recognition process.

In this chapter we present several simple techniques for improving the power of the language model. First, we explore the effect of increasing the vocabulary size on recognition accuracy in an unlimited vocabulary task. Second, we consider ways to model the differences between the language model training text and the way people actually speak. And third, we show that simply increasing the amount of language model training helps significantly.

4.1 The WSJ Corpus

The November 1993 ARPA Continuous Speech Recognition (CSR) evaluations were based on speech and language taken from the Wall Street Journal (WSJ). The standard language model training text was estimated from about 35 million words of text extracted from the WSJ from 1987 to 1989. The text was normalized (preprocessed) with a model for what words people use to read open text. For example, "$234.56" was *always* assumed to be read as "two hundred thirty four dollars and fifty six cents". "March 13" was always normalized as "March thirteenth" – not "March the thirteenth" nor "March thirteen". And so on.

The original processed text contains about 160,000 unique words. However, many of these are due to misspellings. Therefore, the test corpus was limited to those sentences that consisted only of the most likely 64,000 words. While this vocabulary is still quite large, it has two beneficial effects. First, it greatly reduces the number of misspellings in the texts. Second, it allows implementations to use 2-byte data fields to represent the words rather than having to use 4 bytes.

The "standard" recognition vocabulary was defined as the most likely 20,000 words in the corpus. Then, the standard language model was defined as a trigram language model estimated specifically for these 20K words. This standard model, provided by Lincoln Laboratory, was to be used for the controlled portion of the recognition tests. In addition, participants were encouraged to generate an improved language model by any means (other than examining the test data).

4.2 Recognition Lexicon

Typically, we find that over 2% of the word occurrences in a development set are not included in the standard 20K-word vocabulary. Naturally, words that are not in the vocabulary cannot be recognized accurately. (At best, we might try to detect that there is one or more unknown words at some point in a sentence, and then attempt to recognize the phoneme sequence, and then guess a possible letter sequence for this phoneme sequence. Unfortunately, in English, even if we could recognize the phonemes perfectly, there are many valid ways to spell a particular phoneme sequence.) However, in addition to a word not being recognized, we often see that one or two words adjacent to the OOV word are also misrecognized. This is because the recognition, in choosing a word in its vocabulary, also now has the wrong context for the following or preceding words. In general, we find that the word error rate increases by about 1.5 to 2 times the number of OOV words.

One simple way to decrease the percentage of OOV words is to increase the vocabulary size. But which words should be added? The obvious solution is to add words in order of their relative frequency within the full text corpus. There are several problems that might result from this:

1. The vocabulary might have to be extremely large before the OOV rate is reduced significantly.
2. If the word error rate for the vast majority of the words that are already in the smaller vocabulary increased by even a small amount, it might offset any gain obtained from reducing the OOV rate.
3. The language model probabilities for these additional words would be quite low, which might prevent them from being recognized anyway.

We did not have phonetic pronunciations for all of the 64K words. We sent a list of the (approximately 34K) words for which we had no pronunciations to Boston University. They found pronunciations for about half (18K) of the words in their dictionary. When we added these words to our WSJ dictionary, we had a total of 50K words that we could use for recognition.

The following table shows the percentage of OOV words as a function of the vocabulary size. The measurement was done on the WSJ1 Hub1 20K development test which has 2,464 unique words with the total count of 8,227 words.

Vocab.	#OOV	%
19998	187	2.27
28247	85	1.03
35298	39	0.47
40213	14	0.17
41363	12	0.15
48386	1	0.01

Table 4.1: The number of different OOV words and the percentage of OOV words in the test as a function of the vocabulary size

We were somewhat surprised to see that the percentage of OOV words was reduced to only 0.17% when the lexicon included the most likely 40K words – especially given that many of the most likely words were not available because we did not have phonetic

pronunciations for them. Thus, it was not necessary to increase the vocabulary above 40K words.

The second worry was that increasing the vocabulary by too much might increase the word error rate due to the increased number of choices for each of the words that we previous found correctly. So we performed an experiment in which we used the standard 20K language model for the 5K development data. In this case, the increased vocabulary can only increase the error rate. We found, to our surprise, that the error rate increased only slightly, from 8.7% to 9.3%. Therefore, we felt confident that we could increase the vocabulary as needed.

We considered possible explanations for the small increase in error due to a larger vocabulary. We realized that the answer was in the language model. In the first case, when we just increase the vocabulary, the new words also have the same probability in the language model as the old words. However, in this case, all the new words that were added had lower probabilities (at least for the unigram model) than the existing words. Let us consider two possibilities that we would not falsely substitute a new word for an old one. If the new word were acoustically similar to one of the words in the test, and therefore similar to a word in the original vocabulary, then the word would be correctly recognized because the original word would always have a higher language model probability. If, on the other hand, the new word were acoustically very different from the word being spoken, then we might expect that our acoustic models would prevent the new word from being chosen over the old word. While the argument makes some sense, we did not expect the loss for increasing the vocabulary from 5K words to 20K words to be so small.

Finally, the third question is whether the new words would be recognized when they did occur, since (as mentioned above) their language model probabilities were generally low. In fact, we found that, even though the error rate for these new words was higher than for the more likely words, we were still able to recognize about 50% to 70% of them correctly, presumably based largely on the acoustic model. Thus, the net effect of this was to reduce the word error rate by about 1% to 1.5%, absolute.

4.3 Modeling Spoken Language

Another effect that we worked on was the difference between the processed text, as defined by the preprocessor, and the words that people actually used when reading WSJ text. In the pilot WSJ corpus, the subjects were prompted with texts that had already been "normalized", so that there was no ambiguity about how to read a sentence. However, in the WSJ1 corpus, subjects were instructed to read the original texts and to say whatever seemed most

appropriate to them. Since the WSJ1 prompting texts were not normalized to deterministic word sequences, subjects showed considerable variability in their reading of the prompting text.

However, the standard language model was derived from the normalized text produced by the preprocessor. This resulted in a mismatch between the language model and the actual word sequences that were spoken. While the preprocessor was quite good at predicting what people said most of the time, there were several cases where people used different words than predicted. For example, the preprocessor predicted that strings like "$234" would be read as "two hundred thirty four dollars". But in fact, most people read this as "two hundred AND thirty four dollars". For another extreme example, the preprocessor's prediction of "10.4" was "ten point four", but the subject (in the WSJ1 development data) read this as "ten and four tenths". There were many other similar examples.

The standard model for the tests was the "nonverbalized punctuation" (NVP) model, which assumes that the readers never speak any of the punctuation words. The other model that had been defined was the "verbalized punctuation" (VP) model, which assumed that *all* of the punctuation was read out loud. This year, the subjects were instructed that they were free to read the punctuation out loud or not, in whatever way they feel most comfortable. It turns out that people didn't verbalize most punctuation. However, they regularly verbalized quotation marks in many different ways that were all different than the ways predicted by the standard preprocessor.

There were also several words that were read differently by subjects. For example, subjects pronounced abbreviations like, "CORP." and "INC.". While the preprocessor assumed that all abbreviations would be read as full words.

We used two methods to model the ways people actually read text. The simpler approach was to include the text of the acoustic training data in the language model training. That is, we simply added the 37K sentence transcriptions from the acoustic training to the 2M sentences of training text. The advantage of this method is that it modeled what people actually said. The system was definitely more likely to recognize words or sequences that were previously impossible. The problem with this method was that the amount of transcribed speech was quite small (about 50 times smaller) compared to the original training text. We tried repeating the transcriptions several times, but we found that the effect was not as strong as we would like.

A more powerful approach was to simulate the effects of the different word choices by simple rules which were applied to all of the 35M words of language training text. We chose to use the following rules:

Preprocessed Text	Simulated Text
HUNDRED **[number]**	HUNDRED AND **[number]**
ONE HUNDRED	A HUNDRED
ONE DOLLAR	A DOLLAR
ZERO POINT **[number]**	POINT **[number]**
AND ONE HALF	AND A HALF
AND ONE QUARTER	AND A QUARTER

Thus, for example, if the sentence consists of the pattern "hundred twenty", we repeated the same sentence with "hundred AND twenty".

The result was that about one fifth of the sentences in the original corpus had some change reflecting a difference in the way subjects read the original text. Thus, this was equivalent in weight to an equal amount of training text to the original text.

We found that this preprocessing of the text was sufficient to cover most of those cases where the readers said things differently than the predictions. The recognition results showed that the system now usually recognized the new word sequences and abbreviations correctly.

4.4 Increasing the Language Model Training

While 35M words may seem like a lot of data, it is not enough to cover all of the trigrams that are likely to occur in the testing data. So we considered other sources for additional language modeling text. The only easily accessible data available was an additional 3 years (from 1990-1992) of WSJ data from the TIPSTER corpus produced by the Linguistic Data Consortium (LDC).

However, there were two problems with using this data. First, since the test data was known to come from 1987-1989, we were concerned that this might actually hurt performance due to some differences in the topics during that 3-year period. Second, this text had not been normalized with the preprocessor and we did not have available to us the preprocessor that was used to transform the raw text into word sequences.

We decided to use the new text with minimal processing. The text was filtered to remove all tables, captions, numbers, etc. We replaced each initial example of double-quote (") with "QUOTE and the matching token with "UNQUOTE or "ENDQUOTE, which were the most common ways these words were said. No other changes were made. One benefit of this was that abbreviations were left as they appeared in the text rather than expanded. Any numbers, dates, dollar amounts, etc, were just considered "unknown" words, and did not contribute to

the training. We assumed that we had sufficient examples of numbers in the original text.

We found that adding this additional language training data reduced the error by about 7% of the error, indicating that the original 35 million words was not sufficient for the models we were using. Thus, the addition of plain text, even though it was from a different three years, and had many gaps due to apparent unknown words, still improved the recognition accuracy considerably.

4.5 Results

The following table shows the benefit of the enlarged 40K lexicon and the enhanced language model training on the OOV rate and the word error for the development test and the evaluation test.

	% OOV		% Word Error	
Test Set	20K	40K	20K	40K
Development	2.27	0.17	16.4	12.9
Evaluation	1.83	0.23	14.2	12.2

Surprisingly, the addition of three year's LM training (from a period post-dating the test data) improved performance on the utterances that were completely inside the vocabulary. Evidently, even the common trigrams are poorly trained with only the 35 million word WSJ0 corpus. Overall, our modifications to the lexicon and grammar training reduced the word error by 14–22%.

4.6 Spontaneous Dictation

Another area we investigated was spontaneous dictation. The subjects were primarily former or practicing journalists with some experience at dictation. They were instructed to dictate general and financial news stories that would be appropriate for a newspaper like WSJ. In general, the journalists chose topics of recent interest. This meant that the original language model was often out of date for the subject. As a result, the percentage of OOV words increased (to about 4%), and the language model taken from WSJ text was less appropriate.

The OOV words in the spontaneous data were more likely to be proper nouns from recent

events that were not covered by the LM training material. To counter this, we added all (1,028) of the new words that were found in the spontaneous portion of the acoustic training data in WSJ1. This mostly included topical names (e.g., Hillary Rodham, NAFTA, etc.).

In order to account for some of the differences between the read text and the spontaneous text, and to have language model probabilities for the new words, we added the training transcriptions of the spontaneous dictation (about 8K sentences) to the LM training as well.

New weights for the new language model, HMM, and Segmental Neural Network were all optimized on spontaneous development test data. The table below shows that the OOV remains near 1% even after the enlargement to a 41K lexicon.

	% OOV			% Word Error	
Test Set	20K	40K	41K	20K	41K
Development	2.9	1.4	0.8	–	21.7
Evaluation	4.8	1.9	1.5	24.7	19.1

As can be seen, increasing the vocabulary size from 20K to 40K significantly reduced the OOV rate. It is important to point out that in this case, we did not have the benefit of a word frequency list for spontaneous speech, and that the source of speech had an unlimited vocabulary. So the reduction in OOV rate is certainly a fair – if not pessimistic – estimate of the real benefit from increasing the vocabulary. Adding the few new words observed in the spontaneous speech also helped somewhat, but not nearly as much. The sample of only 8,000 sentences is clearly not sufficient to find all the new words that people might use. Presumably, if the sample of spontaneous speech were large enough to derive word frequencies, then we could choose a much better list of 40K words with a lower OOV rate.

Overall, the 41K trigram reduces the word error by 23% over the 20K standard trigram on the November '93 CSR S9 evaluation test. We estimate that more than half of this gain was due to the decreased percentage of OOV words, and the remainder was due to the increased language model training, including specific examples of spontaneous dictation.

4.7 Conclusions

We found the following interesting results:

- Expanding the vocabulary with less frequent words does not substantially increase the word error on words already in the vocabulary, but does eliminate many errors due to OOV words.
- Doubling the amount of language model training text improves the language model, even though the text comes from different years than the test, and even though the text was not preprocessed into proper lexical forms.
- It is possible to improve the quality of the language modeling text by modeling the differences between the predicted reading style and some examples of actual transcriptions.
- Increasing the vocabulary size and language training had a bigger effect on spontaneous speech than it did for read speech.

Chapter 5

Dealing with Spontaneous Speech

In this chapter we describe the work that we performed on making speech recognition work well on spontaneous speech. Spontaneous speech differs from read speech primarily in the style of speaking, with frequent disfluencies of different types. The Air Travel Information Service (ATIS) domain offers an opportunity to explore techniques for dealing effectively with spontaneous speech.

The problem of understanding goal-directed spontaneous speech is harder than recognizing and understanding read text, due to greater variety in the speech and language produced. We have made several modifications to our speech recognition and understanding methods to deal with these variabilities. The speech recognition uses a novel multipass search strategy that allows great flexibility and efficiency in the application of powerful knowledge sources. The result is a very usable system for domains of moderate complexity.

More details on the specific techniques, the ATIS corpus, and the results can be found in the papers presented at the 1992 ARPA Workshop on Speech and Natural Language [16, 17, 18, 19].

5.1 The ATIS Domain and Corpus

The Air Travel Information Service (ATIS) is a system for getting information about flights. The information contained in the database is similar to that found in the Official Airline Guide (OAG) but is for a small number of cities. The ATIS corpus consists of spoken queries by a large number of users who were trying to solve travel related problems. The ATIS2 training corpus consists of 12,214 spontaneous utterances from 349 subjects who were

using simulated or real speech understanding systems in order to obtain realistic speech and language. The data originated from 5 collection sites using a variety of strategies for eliciting and capturing spontaneous queries from the subjects [19].

Each sentence in the corpus was classified as class A (self contained meaning), class D (referring to some previous sentence), or class X (impossible to answer for a variety of reasons). The speech recognition systems were tested on all three classes, although the results for classes A and D were given more importance. The natural language system and combined speech understanding systems were scored only on classes A and D, although they were presented with all of the test sentences in their original order.

The Feb '92 and Nov '92 evaluation test sets had 971 and 967 sentences, respectively, from 37 and 35 speakers with an equal number of sentences from all 5 sites. For both test sets, about 43% of the sentences were class A, 27% were class D, and 30% were class X. The recognition mode was speaker-independent – the test speakers were not in the training set and every sentence was treated independently.

5.1.1 New Extensions for Spontaneous Speech

Spontaneous queries spoken in a problem-solving dialog exhibit a wide variety of disfluencies. There were three very frequent effects that we attempted to solve – excessively long segments of waveform with no speech, poorly transcribed training utterances, and a variety of nonspeech sounds produced by the user.

The long segments of silence were due in part to hesitations as the speakers posed questions to the system and in part to the unpredictable methods employed for endpointing the waveforms, i.e., manual segmentation via push-to-talk and push-hold signals from the user. When background noise is present, the HMM is not a particularly reliable discriminator of speech vs. silence, and many insertion errors result. We chose to find and truncate long regions of nonspeech with a very reliable energy-based speech detector that can deal with noise bursts near the speech. The speech detector uses several simple adaptive SNR-dependent detection thresholds. We eliminated long periods of background with a heuristic energy-based speech detector. But typically, there are many untranscribed short segments of background silence remaining in the waveforms after truncating the long ones, These are not marked in the sentence transcriptions unless transcribers listen carefully to all of the training speech, but they measurably degrade the performance gain usually derived from using cross-word-boundary triphone HMMs. We mark the missing silence locations automatically by running the recognizer on the training data constrained to the correct word sequence, but allowing optional silence between words. Then we retrained the model using the output of the recognizer as *corrected* transcriptions.

Spontaneous data from naive speakers has a large number and variety of nonspeech events, such as pause fillers (um's and uh's), throat clearings, coughs, laughter, and heavy breath noise. We attempted to model a dozen broad classes of nonspeech sounds that were both prominent and numerous. However, when we allowed the decoder to find nonspeech models between words, there were more false detections than correct ones. Because our silence model had little difficulty dealing with breath noises, lip smacks, and other noises, our best results were achieved by making the nonspeech models very unlikely in the grammar.

5.1.2 Forward-Backward N-best Search Strategy

The BYBLOS speech recognition system uses a novel multi-pass search strategy designed to use progressively more detailed models on a correspondingly reduced search space. It produces an ordered list of the N top-scoring hypotheses which is then reordered by several detailed knowledge sources. This N-best strategy [21, 22] permits the use of otherwise computationally prohibitive models by greatly reducing the search space to a few (N=20-100) word sequences. It has enabled us to use cross-word-boundary triphone models and trigram language models with ease. The N-best list is also a robust interface between speech and natural language that provides a way to recover from speech errors.

We use a 4-pass approach to produce the N-best lists for natural language processing.

1. A forward pass with a bigram grammar and discrete HMM models saves the top word-ending scores and times [5].
2. A fast time-synchronous backward pass produces an initial N-best list using the Word-Dependent N-best algorithm[14].
3. Each of the N hypotheses is rescored with cross-word-boundary triphones and semi-continuous density HMMs.
4. The N-best list is rescored with a trigram grammar.

Each utterance is quantized and decoded three times, once with each gender-dependent model and once with a gender-independent model. (In the Feb '92 test we did not use the gender-independent model.) For each utterance, the N-best list with the highest top-1 hypothesis score is chosen. The top choice in the final list constitutes the speech recognition results reported below. Then the entire list is passed to the language understanding component for interpretation.

5.1.3 Training Conditions

We have run evaluations on the ATIS corpus on four separate occasions: Feb91, Feb92, Nov92, and Nov93. For the first test, in Feb91, we only had a very small amount of training speech and language available. Thus, the word error rates for BYBLOS (which were the lowest in the community) were around 16%. On the remaining tests, there was more representative training available, resulting in much lower error rates.

Since there were several tests, below, we provide separate statistics for the three remaining tests [Feb92,Nov92,Nov93] in the form [n1–n2–n3]. We used speech data from the ATIS2 subcorpus exclusively to train the parameters of the acoustic model. However, we filtered the training data for quality in several ways. We removed from the training any utterances that were marked as truncated, containing a word fragment, or containing rare nonspeech events. Our forward-backward training program also automatically rejects any input that fails to align properly, thereby discarding many sentences with incorrect transcriptions. These steps removed about 1,200 utterances from consideration. After holding out a development test set of [890–971–969] sentences, we were left with a total of [7,670–10,925–20,000] utterances for training the HMMs. Since we train gender-dependent models, the training was further divided for the female speakers and for the males.

The recognition lexicon contained [1881–1830–2600] words derived from the training corpus and all the words and natural extensions from the ATIS application database. The lexicon used for recognition was initialized by including all words observed in the complete grammar training texts. This had the side-effect of including the entire development test set within the vocabulary. Common closed-class words such as days of the week, months, numbers, plane types, etc., were completed by hand. Similarly, we included derivations (mostly plurals and possessives) of many open-class words in the domain. We also added about 400 concatenated word tokens for commonly occurring sequences such as WASHINGTON_D_C, and D_C_TEN. On all of the tests, only about 0.5% of the words in the test set were not in the lexicon.

For statistical language model training we used all available [14,500–17,313–29,000] sentence texts from ATIS0, ATIS1, and ATIS2 (excluding the development test sentences from the language model training during the development phase). We estimated the parameters of our statistical bigram and trigram grammars using a new backing-off procedure[23]. The n-grams were computed on [1054–1090–1400] semantic word classes in order to share the very sparse training (most words remained singletons in their class).

5.1.4 Speech Recognition Results

Table 5.1 shows the official results for BYBLOS on this evaluation, broken down by utterance class. We also show the average perplexity of the bigram and trigram language models as measured on the evaluation test sets (ignoring out-of-vocabulary words).

Sentence Class	Bigram Perplex	Trigram Perplex	Feb92–Nov92–Nov93 % Word Errors
A+D	17	12	6.2–4.3–3.3
A+D+X	20	15	9.4–7.6–4.4
A	15	10	5.8–4.0–3.0)
D	20	14	7.0–4.8–4.0)
X	35	28	17.2–14.5–8.6)

Table 5.1: Official SPREC results on Feb92, Nov92, and Nov93 test sets.

The word error rate in each category was lower than any other speech system reporting on this data. The recognition performance was well correlated with the measured perplexities. The trigram language model consistently, but rather modestly, reduced perplexity across all three classes. (However, we observed that word error was reduced by 40% on classes A+D with the trigram model.) More striking are the differences between the perplexities of the A and D sentences and the Class X sentences (those which are unevaluable with respect to the database). The error rate dropped dramatically with each successive test. For example, the error on class A+D sentences went from 18% in 1991 and 6% in early 1992 down to 3% in 1993. This drop is due primarily to the availability of appropriate training speech, and somewhat due to improved methods.

The performance on the class X utterances (those which are unevaluable with respect to the database) is markedly worse than either class A or D utterances. In fact, well over half of the speech errors occur on these utterances. Since these utterances are not evaluable by the natural language component, it does not seem profitable to try to improve the speech performance on these utterances for a spoken language system.

The individual speaker results varied widely from 0.0% word error to 23% error with the median at about 3.0%. The female speakers got significantly better results than the male speakers. Further examination showed that the males in this corpus tended to speak faster and stray from the domain more frequently, resulting in utterances with higher perplexity.

The error rates varied significantly depending on the site where the data was collected. A closer examination of the results (for the Nov92 evaluation) showed that this was due to

several effects, including the number of serious spontaneous speech effects, and the amount of training from each site. We found (based on comparison of each error with the detailed transcriptions) that 22% of the word errors were directly caused by spontaneous speech effects. If we removed the errors due to spontaneous effects, then the remaining error rate was found to vary inversely with the square root of the amount of training data, as expected. Thus, the data from MIT, which had the lowest number of spontaneous effects and by far the most training data, resulted in the lowest error rate. As the error rate was reduced due to more training, a higher percentage of the remaining errors were due to speech disfluencies. This problem remains a difficult one.

In Table 5.2 we observe a large variation in overall performance on the class A+D utterances for each segment of the test data originating at a given collection site, as shown in the rightmost column. We believe that most of this variation can be explained by two easily measured factors – amount of training data from the matching site, and the number of errors due to all spontaneous speech effects. The actual number of training utterances

Site	# Utts Training	% Word Error Due To: Modeling Deficiency	Spontaneous Effects	Overall % Word Error
MIT	3700	2.7	0.5	3.2
BBN	1400	4.5	0.8	5.3
CMU	1000	5.3	0.5	5.8
SRI	800	5.7	2.0	7.7
AT&T	800	6.4	4.0	10.4

Table 5.2: BYBLOS performance on February '92 test as a function of originating site (class A+D).

that we used from each site is shown in Table 5.2. The next column shows the word error rate that we attribute to general modeling deficiencies after removing those errors that we judged were due to spontaneous speech effects. The variation due to modeling seems well correlated to the amount of training data available from each site. The numbers show the expected halving of the error rate for a quadrupling of the training data. In particular, we feel that the higher performance on the MIT data can be explained entirely by the increased amount of data from that particular site.

The errors due to spontaneous speech effects in Table 5.2 were counted by matching the output of BYBLOS against the detailed transcriptions. The transcriptions contain specific markings for many spontaneous speech effects including: nonspeech events, word fragments, mispronunciations, emphatic stress, lengthening, and verbal deletions. Any error that

occurred in the immediate vicinity of such a marking was counted as an error due to spontaneous speech. The table shows that the noticeably worse performance on data from SRI and AT&T can be explained by the larger proportion of errors due to spontaneous speech effects. It also shows that errors due to spontaneous speech effects account for only about 22% of the total.

5.2 Real-Time Implementation

A real-time demonstration of the entire spoken language system described above has been implemented. The speech recognition was performed using BBN HARK^{TM}, a commercially available product for continuous speech recognition of medium-sized vocabularies (about 2,000 words). HARK stands for High Accuracy Recognition Kit. HARK^{TM} is based on similar techniques to BYBLOS, but does not include the most advanced and expensive techniques. It can run in real-time entirely in software on a workstation with a built-in A/D converter (e.g., SGI Indigo, SUN Sparc, or HP715) without any additional hardware.

The speech recognition displays an initial answer as soon as the user stops speaking, and a refined (rescored) answer within 1–2 seconds. The natural language system chooses one of the N-best answers, interprets it, and computes and displays the answers, along with a paraphrase of the query so the user can verify what question the system answered. The total response cycle is typically 3–4 seconds, making the system feel extremely responsive. The error rates for knowledgeable interactive users appears to be much lower than those reported above for naive noninteractive users.

5.3 Summary

We have described the techniques and results for the BYBLOS speech recognition system when used to recognize spontaneous speech as part of a spoken language understanding system. BYBLOS is connected to the understanding component via an N-best interface, which is a modular and efficient way to combine multiple knowledge sources at all levels within the system. In addition, the N-best strategy was shown to be useful within the speech recognition system as a means of applying expensive knowledge sources, such as cross-word acoustic models and trigram language models. For the Class A+D subset of the November '93 DARPA test the official BYBLOS speech recognition results were 3.3% word error.

Finally, the entire system has been implemented to run in real time on a standard workstation without the need for any additional hardware.

Chapter 6

Robust Speech Recognition

In the next chapter, we discuss experiments and methods developed specifically for dealing with testing conditions that are markedly different from the conditions in the training data.

6.1 Experiments in Robust Speech Recognition

Our recent work was focused in four specific problem areas and tested in the 1993 ARPA CSR evaluation. A common thread running through these experiments is that the test condition exposes the recognizer to phenomena not observed in the training data. In all of the following experiments, we used the SI-37 corpus for acoustic training of the HMMs. This corpus consists of 12 speakers from WSJ0 plus 25 speakers from WSJ1. All of this data is read speech from Wall Street Journal texts, by native speakers of American English, and from the Sennheiser channel only. These experiments were supported by the new "Hub and Spoke" testing paradigm proposed by BBN and adopted by the 4C committee (Continuous Speech Corpus Collection Committee). This paradigm facilitates carefully controlled experiments on many specific problem areas of interest.

Here we investigated

1. spoken language effects due to subject variability and spontaneous dictation,
2. non-native speakers of the language, and
3. new microphones not used in training.

6.1.1 Very Large Vocabulary SI Recognition

In this work we attempted to go beyond the standard 20K-word open-vocabulary grammar created by Doug Paul at Lincoln Laboratory.

The test data, which was collected using raw text versions of the prompts used in WSJ0, has a vocabulary of 64K-words. The lexicon of the standard 20K open grammar was defined by taking the 20K most frequent words measured over the LM training data. This training data consists of normalized texts from 1987-1989 WSJ, or about 35 million words.

We found that about 2.5% of a development set was not included in the 20K open vocabulary. We looked at the OOV rate as a function of vocabulary size on a development test set and found that we could reduce the OOV rate to about 0.2% by including the 40K most frequent words in the lexicon.

We added, to the LM training, an additional 3 year's of WSJ data from the TIPSTER corpus produced by the Linguistic Data Consortium (LDC). This text was filtered to remove all tables, captions, numbers, etc.

We modeled spoken language effects that we observed in the acoustic training data. Since the WSJ1 prompting texts were not normalized to deterministic word sequences, subjects showed considerable variability in their reading of the prompting text. For example, while the standard language model would not allow it, subjects often inserted the word, "AND", in the phrase, "one hundred AND sixty". Additional variability was introduced by subjects pronouncing abbreviations like, "CORP." and "INC." or verbalizing the punctuation. We modeled these effects by manipulating the normalized texts from WSJ0 and adding them to the LM training. The table below shows the effect of expanding the vocabulary on OOV

	% OOV		% Word Error	
Test Set	20K	40K	20K	40K
Development	2.5	0.2	16.4	12.9
Evaluation	1.7	0.2	14.3	12.3

and word error rate. Surprisingly, the addition of 3 year's LM training (from a period post-dating the test data) improved performance on the utterances that were completely inside the vocabulary. Evidently, even the common trigrams are poorly trained with only the 35 million word WSJ0 corpus. Overall, our modifications to the lexicon and grammar training reduced the word error by 14–17%.

6.1.2 Spontaneous Dictation

Another area we investigated was spontaneous dictation. The subjects were primarily former or practicing journalists with some experience at dictation. They were instructed to dictate general and financial news stories that would be appropriate for a newspaper like WSJ.

The OOV words in the spontaneous data were more likely to be proper nouns from recent events that were not covered by the LM training material. To counter this, we added approximately 1000 new words that were found in the spontaneous portion of the acoustic training data in WSJ1. This portion numbers about 8K utterances. We added it to the LM training as well. New weights for the new LM, HMM, and SNN were all optimized on spontaneous development test data. The table below shows that the OOV remains near 1%

	% OOV			% Word Error	
Test Set	20K	40K	41K	20K	41K
Development	2.9	1.4	0.8	–	21.7
Evaluation	4.8	1.9	1.5	26.1	20.4

even after the enlargement to a 41K lexicon. Overall, the 41K trigram reduces the word error by 22% over the 20K standard trigram on the November '93 CSR S9 evaluation test.

6.1.3 Adaptation for Outlier Speakers

Another area we investigated was how to repair the SI performance degradation of outlier speakers who are non-native speakers of American English. The speakers were primarily from Europe and Asia, and spoke English as their second or third language.

We proposed to reduce this problem by using a speaker-transformation based on a Probabilistic Spectral Mapping (PSM) that adapts the HMM of each reference speaker in the SI training set, so that it models the target speaker. The adaptation is performed using 40 sentences of adaptation speech that have been collected from every training and test speaker. First, a speaker-dependent codebook is created for the target speaker. We time align the corresponding sentences from each training speaker and the target speaker. The VQ index of the aligned frames are used to create a probabilistic confusion matrix between the target speaker and each training speaker. This matrix was normalized to produce a probabilistic VQ mapping. Then, each discrete pdf for the training speaker was multiplied by the appropriate matrix to produce a model for the target speaker. Each transformed model was used

to recognize the adaptation data from the target speaker in order to determine how well this new model matched the target speaker. Finally the transformed models were averaged with weights that depended on the recognition error rate on the adaptation data. For native speak-

	Recognition Paradigm	
Test Set	SI-37	SA-37
Development	~45	17.3
Evaluation	32.3	14.8

ers, we expect this system to achieve about 7-8% word error averaged over the speakers. The table above shows a 4-6 fold increase in expected error for non-native speakers in SI mode. With PSM adaptation, however, the degradation factor is reduced to 2-2.5 times the SI error.

It is surprising that so much of the degradation due to non-native dialect can be removed by a correction in the spectral space only. Note that we did not modify the phoneme inventory, the pronunciation dictionary, or the language model.

6.1.4 Adaptation to a Known Microphone

In a separate paper [4] we consider the problem of adapting a large corpus of acoustic training data collected on one microphone to a new one, for which a small sample of data (400 utterances) is available for adaptation.

We found that, after adaptation, the word error rate for a boom microphone and a telephone handset were only slightly higher than that of the Sennheiser microphone, even though we used models derived from the Sennheiser channel only.

In this paper, we present several approaches designed to increase the robustness of BYBLOS, the BBN continuous speech, Hidden Markov Model (HMM) recognition system. We address the problem of increased degradation in performance when there is mismatch in the characteristics of the training and the test microphones. First we compare RASTA processing and Cepstrum Mean Subtraction as preprocessing methods, to compensate for unknown channel transfer function effects, when we have no information about the new microphone. Then we introduce a new algorithm that computes a probabilistic transformation from the training microphone codebook to that of a new microphone, given some information about the new microphone. We test this algorithm in supervised mode and, combined with a microphone selection method, in unsupervised mode. We present experimental results which

show that the proposed algorithm combined with cepstrum mean subtraction, improves the recognition accuracy when the system is tested on a microphone with different characteristics than the one on which it was trained.

6.2 Introduction

Interactive speech recognition systems are usually trained on substantial amounts of speech data collected with a high quality close-talking microphone. During recognition, these systems require the same type of microphone to be used in order to achieve their standard accuracy. This is a highly restricting condition for practical applications of speech recognition systems. One can imagine a situation where it would be desirable to use a different microphone for recognition than the one with which the training speech was collected. For example, some users may not want to wear a headmounted microphone. Others may not want to pay for a high quality microphone. Additionally, many applications involve recognition of speech over telephone lines and telephone sets with high variability in quality and characteristics. However, we know that even highly accurate automatic speech recognition systems perform very poorly when they are tested with microphones with different characteristics than the ones, that they were trained on [26, 27]. We could compensate for this degradation in performance either by retraining the HMMs with data collected with the new microphone encountered during the recognition stage, a rather expensive approach for real applications, or by training on a large number of microphones in the hope that the system will obtain the necessary robustness. In this paper we present a different approach by modeling the difference between the test and the training microphone prior to recognition.

We have developed a technique for adaptation to a new microphone based on modifying the continuous densities in a tied-mixture HMM system, using a relatively small amount of stereo training speech. We call this method *Tied-Mixture Normalization (TMN)* and it is presented in Section 3. In section 4 we use this method to address the microphone independence problem, by combining it with a microphone selection algorithm.

Prior to developing the new algorithm, in the context of microphone independence, we evaluated the RASTA algorithm and the *Cepstrum Mean Subtraction*, two simple cepstrum preprocessing methods, that try to alleviate the effect of linear spectral distortion of recorded speech.

6.3 Cepstrum Preprocessing

Microphones distort the speech signal mainly into two distinct ways. First, they allow different levels of ambient noise that account for an additive effect in the recording speech and second they act as unknown linear filters, causing a variable spectral tilt that depends on the specific microphone characteristics. The convolutional effect appears as an additive constant both in the log spectrum and the cepstrum domain. The *RelAtive SpecTrAl (RASTA) Processing* [28] aims at removing the influence of an unknown, slowly time varying channel transfer function on the speech features. It smoothes the cepstrum vector with a five-frame averaging window, and also removes the effect of a slowly varying multiplicative filter, by subtracting an estimate of the average cepstrum. In Cepstrum Mean Subtraction we compute the sample mean of the cepstrum vector over the utterance, and then subtract this mean from the cepstrum vector at each frame. In both methods, speech frames are not distinguished from noise frames. The processing is applied to all frames equally.

6.4 Tied Mixture Normalization

In a *Tied-Mixture Hidden Markov Model (HMM)* system [29, 30], speech is represented by an ensemble of Gaussian mixture densities. Every frame of speech that is quantized, is represented by a set of mean vectors and variances that characterize the mixture density codebook. This codebook has been derived from a subset of the training data, therefore it is mostly characteristic of the location and distribution of the training data and the training microphone in the acoustic space. However if the codebook was created with data collected with some other microphone, due to the additive and convolutional effect on speech specific to this new microphone, the data would be distributed differently in the acoustic space and the ensemble of means and variances of the codebook would reflect the characteristics of the new microphone. This is the case of the mismatch in training and testing microphone. Without any compensation, we quantize the test data recorded with the new microphone, using the mixture codebook generated from recordings with the training microphone. This inevitably results in a degradation in performance, since the codebook does not model the test data.

We introduce a new algorithm, called *Tied Mixture Normalization (TMN)* to compute the codebook transformation from the training microphone to the new test microphone. The TMN algorithm requires a relatively small amount of stereo speech adaptation data, recorded simultaneously with the microphone used for training (primary microphone) and the new microphone (secondary microphone). Then using the stereo data we can adapt the existing HMM model to work well on the new testing condition despite the mismatch with

the training conditions.

We assume that we have a tied-mixture densities codebook (set of Gaussians distributions), derived from a subset of the training data that was recorded with the primary microphone. These Gaussian distributions are used as the bases of the tied-mixture distributions. We quantize the adaptation data from the first channel and we label each frame of speech with the index of the most likely Gaussian distribution in the tied-mixture codebook. Since there is an one-to-one correspondence between data of the first and second channel we use the VQ indices of the frames of the data of the first channel to label the corresponding frames of the data of the second channel. Then for each of the VQ clusters, from all the frames of the secondary microphone data with the same VQ label, we compute the sample mean and the sample covariance of the cepstrum vectors that represent a possible shift and scaling of this cluster in the acoustic space

the Gaussian distributions of the codebook and define a normalized codebook transformation to accommodate the secondary microphone. The new Gaussians distributions are then used in conjunction with the mixture weights (sometimes called the discrete probabilities) of the original model.

One of the possible weaknesses of the TMN algorithm is that each cluster of the original codebook is transformed independently of all the others. This assumption goes against our intuition that a codebook transformation due to different microphone characteristics should maintain continuity between adjacent codebook clusters and shift all the clusters in the same general direction. Modeling each codebook cluster independently, we may not estimate the correct transformation due to insufficient or distorted data. To alleviate this problem we use the following approach, originally suggested for speaker adaptation [31]: when the centroid of the $i$the codebook cluster is denoted by m_i and that of the transformed secondary microphone by μ_i, the deviation vector between these two centroids is

$$d_i = \mu_i - m_i \quad i = 1, 2, ..., C \tag{6.1}$$

where C is the size of the codebook. For each cluster centroid c_i, the deviation vectors of all clusters $\{d_i\}$ are summed with weighting factors $\{w_{ik}\}$ to produce the shift vector Δ_i:

$$\Delta_i = (\sum_{k=1}^{C} w_{ik} d_i) / (\sum_{k=1}^{C} w_{ik}) \tag{6.2}$$

The weighting factor w_{ik} is the probability $\{P(m_k|m_i)\}^\alpha$ of centroid m_k of the original codebook to belong to the ith cluster raised to the α power. This weight is a measure of vicinity among clusters and the exponentiation controls the amount of smoothing between the clusters. Finally, the centroid c_i' of the ith cluster of the transformed codebook is:

$$c_i' = c_i + \Delta_i \tag{6.3}$$

Similarly the variances of the clusters of the new codebook are computed as the averaged summations over all sample variances computed in the first implementation of TMN.

6.5 Microphone Independence

We present some work that we initially conducted towards the goal of microphone independence using the TMN algorithm. The identity of the test microphone is not known in microphone independence and the problem becomes much more complicated. In this case, one technique is to estimate a TMN transformation for many different types of microphones and then select one of those transformations.

We had available stereo training data using several microphones that were not used in the test. We grouped the secondary microphones in the training into six broad categories, such as lapel, telephone, omni-directional, directional microphones, and two specific desk-mounted microphones. Then, using the TMN algorithm we estimated a transformed codebook for each of the microphone classes using stereo data from that microphone class and the Sennheiser, being sure that the adaptation data included both male and female speakers.

To select which microphone transformation to use, for each of the seven microphone types (Sennheiser plus six alternate types) we estimated a mixture density consisting of eight Gaussians distributions. Then, given a sentence from an unknown microphone, we computed the probability of the data being produced by each of the seven mixture densities. The one with the highest likelihood was chosen, and we then used the transformed codebook corresponding to the chosen microphone type. We found that on development data this microphone selection algorithm was correct about 98% of the time, and had the desirable property that it it *never* misclassified the Sennheiser data.

6.6 Description of Experiments

All of the experiments that will be described were performed using the BBN BYBLOS speech recognition system [20].

6.6.1 Cepstrum Preprocessing Results

The system was trained on the Wall Street Journal (WSJ) pilot corpus (WSJ0) which contains 12 hours of training speech recorded from 12 speakers and was tested on the 5K-word development test using a bigram language model. The RASTA preprocessing and the Cepstrum Mean Subtraction were tested on the verbalized punctuation (VP) 5K-word test set. Every test utterance was recorded simultaneously on the same microphone used in the training (a high-quality noise-canceling Sennheiser microphone) and on some other microphone which was not known, but which ranged from an omni-directional boom-mounted microphone or table-mounted microphone, a lapel microphone, or a speaker-phone. We present the error rates for the baseline system which uses the normal mel-cepstral values and for the two preprocessing methods in Table 6.1. The results show that the word error rate increases by a factor of three when the microphone is changed radically. The RASTA algorithm reduced the degradation to a factor of 2.3, while degrading the performance on the Sennheiser microphone just slightly. The mean subtraction also reduced the degradation, but did not degrade the performance on the training microphone.

	Sennheiser	Secondary-Mic
Baseline System	12.0%	37.7%
Rasta Preprocessing	12.5%	27.8%
Cepstral Mean Subtraction	11.8%	27.2%

Table 6.1: Comparison of word error rate among the baseline BYBLOS system and two cepstrum preprocessing techniques

6.6.2 Unknown microphone adaptation results

We compare the performance of the baseline system with no preprocessing, with the same system using cepstrum mean subtraction and one that uses the combination of mean subtraction and the microphone adaptation strategy we described in section 4 for the case of unknown microphones. Using the same training configuration, the recognition was performed on the 5K-word non-verbalized punctuation (NVP) development test set using a bigram language model. The test contained speech collected with four microphones which we try to model with the generic microphone types transformations we described in section 6.5.

Again, the cepstral mean subtraction reduced the degradation somewhat. The TMN algorithm with microphone selection reduced the error rate by 30% relative to the cepstral mean subtraction.

	Sennheiser	Secondary-Mic
Baseline System	11.6%	40.2%
Cepstral Mean Subtraction	11.3%	32.4%
Tied-Mixture Normalization	11.3%	21.3%

Table 6.2: Comparison of word error rate for the task of microphone independence

6.6.3 Known Microphone Adaptation Results

Finally we describe our latest results on the task of microphone adaptation to a known microphone. The configuration of the recognition system is improved mainly by adding more training data. We used 62 hours of training speech from the WSJ0 and WSJ1 corpora, collected from 37 speakers, with a Sennheiser close-talking microphone. Cepstrum Mean Subtraction is used as a standard feature of our recognition system front-end. The recognition is done using trigram language models. The test data comes from the development and evaluation sets of the WSJ1 corpus and consists of stereo recordings with Sennheiser microphone and an Audio-Technica 853a directional stand-mounted microphone or a telephone handset over external telephone lines. Adaptation data was supplied separately consisting a total of 800 stereo recorded utterances from 10 speakers; 400 sentences recorded simultaneously with the Sennheiser and the Audio-Technica and 400 sentences recorded with the Sennheiser and the telephone handset. The telephone handset differs radically from the other two microphones, having the main characteristic of allowing a much narrower band of frequencies than the others. Therefore we chose to bandlimit the Sennheiser training data between 300-3300 Hz to create new bandlimited phonetic word models, prior to applying any adaptation scheme. We also bandlimited the stereo adaptation data for the telephone handset. After the bandlimiting processing, we applied the TMN algorithm to both sets of adaptation data to generate the codebook transformations specific to the new microphones. During testing, the telephone data is bandlimited first, and data collected with both microphones is quantized using the corresponding transformed codebook. In Tables 6.3 and 6.4 we show the error rates of the baseline system when tested on the Sennheiser part of the data, and when it is tested on either of the secondary microphones. The mismatch between the Audio-Technica and the Sennheiser microphone does not cause a serious degradation compared with the degradation in error rate due to the mismatch between the telephone handset and the Sennheiser, which is severe. Furthermore, we see that the TMN adaptation reduces the degradation in both cases. Combined with bandlimiting the data, it has a significant effect on the telephone data reducing the error rate by a factor of 2.3.

	Dev	Eval
Sennheiser	8.3%	7.9%
Audio-Technica with no adaptation	-	10.6%
Audio-Technica with TMN adaptation	9.0%	9.6%

Table 6.3: Comparison of word error rate for microphone adaptation using the Sennheiser or the Audio-Technica microphone

	Dev	Eval
Sennheiser	8.9%	8.7%
Telephone handset with no adaptation	-	29.5%
Telephone handset with Bandlimiting and TMN adaptation	12.7%	12.8%

Table 6.4: Comparison of word error rate for microphone adaptation using the Sennheiser or the Telephone handset microphone

6.7 Conclusions

We described and evaluated three different methods to improve the robustness of the BYBLOS recognition system with respect to mismatches between the training and test microphone. First, we implemented the *Cepstral Mean Subtraction*, a very simple cepstrum preprocessing technique. This method improves the recognition accuracy of the base system when a new test microphone is used, while it does not degrade the baseline performance when the system is tested on the training microphone. As a result we incorporated this method in the BYBLOS front-end as a standard feature.

Next we presented the *Tied Mixture Normalization* algorithm, that computes a probabilistic transformation to map the codebook created from data collected with the training microphone to a codebook that models the data collected with the new test microphone. The algorithm uses only small amounts of stereo adaptation data to compute the transformation. We saw a significant improvement in performance by using this algorithm to adapt to two new microphones different from the training one.

Finally we developed a microphone selection algorithm to enable the Tied Mixture Normalization to do unsupervised adaptation. In this case the method is used with adaptation data that come from generic microphone classes. The microphone selection algorithm is used

to select the most like codebook transformation. Our experiments showed that the combination of microphone selection with TMN gives a significant improvement in accuracy. Very important, that the microphone selection algorithm does not misclassify the training microphone so this method does not degrade the recognition accuracy when the same microphone is used for training and testing.

6.8 Summary

1. We were able to reduce the word error rate obtained by nonnative speakers by more than a factor of two, using the supervised PSM technique with 40 adaptation sentences.

2. Increasing the vocabulary size and language training had a bigger effect on spontaneous speech than it did for read speech.

3. Testing with a microphone that is different from the one used during training does not have to be a serious problem. If the microphone is known, we can use an appropriate transformation on the models. If the microphone is unknown, we can determine an appropriate transformation from the overall characteristics of the speech.

Bibliography

[1] A. Asadi, R. Schwartz, and J. Makhoul, "Automatic Modeling for Adding New Words to a Large-Vocabulary Continuous Speech Recognition System," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Toronto, Canada, May 1991, pp. 305-308.

[2] F. Kubala and R. Schwartz, "A New Paradigm for Speaker-Independent Training," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Toronto, Canada, May 1991, pp. 833-836.

[3] F. Kubala, A. Anastasakos, J. Makhoul, L. Nguyen, R. Schwartz, and G. Zavaliagkos, "Comparative Experiments on Large Vocabulary Speech Recognition," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Adelaide, Australia, April, 1994, pp. 561-564.

[4] A. Anastasakos, F. Kubala, J. Makhoul, and R. Schwartz, "Adaptation to New Microphones Using Tied-Mixture Normalization," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Adelaide, Australia, April, 1994, pp. 433-436

[5] S. Austin, R. Schwartz, and P. Placeway, "The Forward-Backward Search Algorithm," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Toronto, Canada, May 1991, pp. 697-700.

[6] M. Bates, R. Bobrow, P. Fung, R. Ingria, F. Kubala, J. Makhoul, L. Nguyen, R. Schwartz, and D. Stallard, "The BBN/HARC Spoken Language Understanding System," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, MN, April 1993, pp. 111-114, Vol. II.

[7] H-W. Hon, "Vocabulary-Independent Speech Recognition: The VOCIND System," Doctoral Thesis, CMU, March 1992.

[8] M-Y. Hwang and X. Huang, "Subphonetic Modeling with Markov States – Senone," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, March 1992, pp. 33-36, Vol. I.

[9] M-Y. Hwang, X. Huang, and F. Alleva, "Predicting Unseen Triphones with Senones," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, MN, April 1993, pp. 311-314, Vol. II.

[10] H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Progressive-Search Algorithms for Large-Vocabulary Speech Recognition," *Proc. of the ARPA Workshop on Human Language Technology*, March 1993.

[11] M. Ostendorf, A. Kannan, S. Austin, O. Kimball, R. Schwartz, and J.R. Rohlicek, "Integration of Diverse Recognition Methodologies Through Reevaluation of N-Best Sentence Hypotheses," *Proc. of the DARPA Speech and Natural Language Workshop*, Morgan Kaufmann Publishers, Feb. 1991, pp. 83-87.

[12] V. Digalakis and H. Murveit, "Genones: Optimizing the Degree of Mixture Tying in a Large Vocabulary Hidden Markov Model Based Speech Recognizer," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Adelaide, Australia, April, 1994, pp. 537-540.

[13] R. Schwartz, O. Kimball, F. Kubala, M-W. Feng, Y-L. Chow, C. Barry, and J. Makhoul, "Robust Smoothing Methods for Discrete Hidden Markov Models," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Glasgow, Scotland, May 1989, pp. 548-551.

[14] R. Schwartz and S. Austin, "A Comparison Of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing* Toronto, Canada, May 1991, pp. 701-704.

[15] Y. Zhao, J. Makhoul, R. Schwartz, and G. Zavaliagkos, "Segmental Neural Net Optimization for Continuous Speech Recognition," Presented at the *Neural Information Processing Conference*, Denver, CO, November, 1993.

[16] F. Kubala, C. Barry, M. Bates, R. Bobrow, P. Fung, R. Ingria, J. Makhoul, L. Nguyen, R. Schwartz, and D. Stallard, "BBN BYBLOS and HARC February 1992 ATIS Benchmark Results," *Proc. of the DARPA Speech and Natural Language Workshop*, Harriman, NY, Morgan Kaufmann Publishers, Feb. 1992, pp. 72-77.

[17] R. Bobrow and D. Stallard, "Fragment Processing in the DELPHI System," *Proc. of the DARPA Speech and Natural Language Workshop*, Harriman, NY, Morgan Kaufmann Publishers, Feb. 1992.

[18] R. Bobrow, R. Ingria, and D. Stallard, "Syntactic/Semantic Coupling in the DELPHI System," *Proc. of the DARPA Speech and Natural Language Workshop*, Harriman, NY, Morgan Kaufmann Publishers, Feb. 1992.

[19] MADCOW, "Multi-Site Data Collection for a Spoken Language Corpus," *Proc. of the DARPA Speech and Natural Language Workshop*, Harriman, NY, Morgan Kaufmann Publishers, Feb. 1992.

[20] Y. Chow, M. Dunham, O. Kimball, M. Krasner, G.F. Kubala, J. Makhoul, P. Price, S. Roucos, and R. Schwartz, "BYBLOS: The BBN Continuous Speech Recognition System," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Dallas, Texas, April 6-9, 1987, pp. 89-92.

[21] Y-L. Chow, and R.M. Schwartz, "The N-Best Algorithm: An Efficient Procedure for Finding Top N Sentence Hypotheses," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Albuquerque, NM S2.12, pp. 81-84.

[22] R. Schwartz, S. Austin, F. Kubala, and J. Makhoul, "New Uses for the N-Best Sentence Hypotheses Within the BYBLOS Speech Recognition System," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing* , San Francisco, CA, March 23-26, 1992, pp. I.1-I.4.

[23] P. Placeway, R. Schwartz, P. Fung, and L. Nguyen, "The Estimation of Powerful Language Models from Small and Large Corpora," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing* , Minneapolis, MN, April 27-30, 1993, pp. II-33-36.

[24] D. Stallard, "Unification-Based Semantic Interpretation in the BBN Spoken Language System," *Proc. of the DARPA Speech and Natural Language Workshop*, Morgan Kaufmann Publishers, Oct. 1989, pp. 39-46.

[25] R. Bobrow, "Statistical Agenda Parsing," *Proc. of the DARPA Speech and Natural Language Workshop*, Morgan Kaufmann Publishers, Feb. 1991, pp. 222–224.

[26] A. Acero and R.M. Stern, "Environmental Robustness in Automatic Speech Recognition," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Dallas, Texas, April 6-9, 1987, pp. 849-852.

[27] A. Erell and M. Weintraub, "Estimation Using Log-Spectral-Distance Criterion For Noise-Robust Speech Recognition," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Dallas, Texas, April 6-9, 1987, pp. 853-856.

[28] H. Hermansky and N. Morgan, "Towards Handling the Acoustic Environment in Spoken Language Processing," *Proc. International Conference in Spoken Language Processing*, 1992, pp. 85-88.

[29] J. Bellegard and D. Nahamoo, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Dec. 1990, Vol. 38, No. 12.

[30] X. Huang, K. Lee and H. Hon, "On Semi-Continuous Hidden Markov Modeling," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Albuquerque, New Mexico, April 3-6, 1990, paper S13.3.

[31] S. Furui, "Unsupervised Speaker Adaptation Method Based on Hierarchical Spectral Clustering," *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Glasgow, Scotland, May 23-26, 1989, pp. 286-289.

[32] F. Kubala, R. Schwartz, and J. Makhoul, "Dialect Normalization through Speaker Adaptation," *IEEE Workshop on Speech Recognition*, Arden House, Harriman, NY, December 1991.

[33] M. Bates, R. Bobrow, P. Fung, R. Ingria, F. Kubala, J. Makhoul, L. Nguyen, R. Schwartz, and D. Stallard, "Design and Performance of HARC, The BBN Spoken Language Understanding System," *Int. Conf. on Speech and Language Processing*, Banff, Canada, October 12-16, 1992, pp. 241-244.

[34] R. Schwartz, A. Anastasakos, F. Kubala, J. Makhoul, L. Nguyen, and G. Zavaliagkos, "Comparative Experiments on Large Vocabulary Speech Recognition," *Proc. ARPA Workshop on Human Language Technology*, Princeton, NJ, March 1993.

[35] L. Nguyen, R. Schwartz, F. Kubala, and P. Placeway, "Search Algorithms for Software-Only Real-Time Recognition with Very Large Vocabularies," *Proc. ARPA Workshop on Human Language Technology*, Princeton, NJ, March 1993.

[36] R. Schwartz, L. Nguyen, F. Kubala, G. Chou, G. Zavaliagkos, and J. Makhoul, "On Using Written Language Training Data for Spoken Language Modeling," *ARPA Human Language Technology Workshop*, Princeton, N.J., March 6-8, 1994, pp. 93-96.

[37] L. Nguyen, R. Schwartz, and J. Makhoul, "Is N-Best Dead?," *ARPA Human Language Technology Workshop*, Princeton, N.J., March 6-8, 1994, pp. 93-96.